Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 5: UML profile for TDL
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.3.10</td>
<td>ComponentInstanceBinding</td>
<td>44</td>
</tr>
<tr>
<td>9.3.11</td>
<td>ActionBehaviour</td>
<td>44</td>
</tr>
<tr>
<td>9.3.12</td>
<td>ActionReference</td>
<td>45</td>
</tr>
<tr>
<td>9.3.13</td>
<td>InlineAction</td>
<td>45</td>
</tr>
<tr>
<td>9.3.14</td>
<td>Assignment</td>
<td>46</td>
</tr>
<tr>
<td>History</td>
<td></td>
<td>47</td>
</tr>
</tbody>
</table>
Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for ETSI members and non-members, and can be found in ETSI SR 000 314: "Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This ETSI Standard (ES) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

The present document is part 5 of a multi-part deliverable. Full details of the entire series can be found in part 1 [1].

Modal verbs terminology

In the present document "shall", "shall not", "should", "should not", "may", "need not", "will", "will not", "can" and "cannot" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"must" and "must not" are NOT allowed in ETSI deliverables except when used in direct citation.
1 Scope

The present document specifies how the concepts of the TDL meta-model [1] are mapped to OMG® UML® to create a UML Profile for TDL, called UP4TDL.

NOTE: OMG® and UML® are the trademarks of OMG (Object Management Group). This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of the products named.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.


NOTE: Available at http://www.omg.org/spec/UML/2.4.1/.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI ES 203 119-1 [1] and the following apply:

UML profile: extension mechanism provided by UML
3.2 Symbols

For the purposes of the present document, the symbols given in [2] apply.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

- OMG: Object Management Group®
- SUT: System Under Test
- TDL: Test Description Language
- UML: Unified Modelling Language®

4 Structure of the UML Profile for TDL

The stereotypes representing concepts from the Foundation section of the TDL meta-model are directly stored in the UP4TDL Profile, while other concepts are stored in various additional included Packages.

![Figure 4.1: Structure of UMLProfile4TDL](image)

The following clauses describe the content of each package. The subclauses describe how the TDL meta-model elements can be mapped to UML.
5 Foundation

5.1 Overview

Most concepts of the Foundation Package are directly mapped to UML meta-classes. Exceptions are:

- TDL::Element: a stereotype is created to allow elements to have Annotations;
- TDL::Annotations and TDL::TestObjective for which there is no equivalent concept in UML.

![Figure 5.1: Foundational language concepts](image)

5.2 Element

Extended UML Meta-Class

- UML::Element

Generalization

None.

Properties

- `<<UP4TDL::Element>>`
- TDL::Element.comment := Computed as the set of Comment whose `annotatedElement` Property contains this element
- TDL::Element.annotation := UP4TDL::Element.annotation : new (derived) property computed as the set of Comment with stereotype Annotation applied whose `annotatedElement` property contains this Element

Constraints

None.

5.3 NamedElement

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::NamedElement
Generalization
None.

Properties
- TDL::NamedElement.qualifiedName := UML::NamedElement.qualifiedName

Constraints
None.

5.4 PackageableElement

Extended UML Meta-Class
- Direct mapping without a stereotype to UML::PackageableElement

Generalization
None.

Properties
None.

Constraints
None.

5.5 Package

Extended UML Meta-Class
- Direct mapping without a stereotype to UML::Package

Generalization
None.

Properties
- TDL::Package.packagedElement := UML::Package.packagedElement
- TDL::Package.import := UML::Package.elementImport
- TDL::Package.nestedPackage := UML::Package.nestedPackage

Constraints
None.

5.6 ElementImport

Extended UML Meta-Class
- Direct mapping without a stereotype to UML::ElementImport
Generalization
None.

Properties
- TDL::ElementImport is resolved in UML by a set of UML::ElementImport
- TDL::ElementImport.importedPackage := UML::ElementImport.importedElement.namespace
- TDL::ElementImport.importedElement := UML::ElementImport.importedElement

Constraints
None.

Model to Model transformation advice
- In UML, an ElementImport can import exactly one element. This implies that for one TDL::ElementImport, the equivalent model in UP4TDL can have several UML::ElementImport

### 5.7 Comment

Extended UML Meta-Class
- Direct mapping without a stereotype to UML::Comment

Generalization
None.

Properties
- TDL::Comment.commentedElement := UML::Comment.annotatedElement
- TDL::Comment.body := UML::Comment.body

Constraints
None.

### 5.8 Annotation

Extended UML Meta-Class
- UML::Comment

Generalization
None.

Properties
- <<UP4TDL::Annotation>>
- TDL::Annotation.key := UP4TDL::Annotation.key (new property)
- TDL::Annotation.value := UP4TDL::Annotation.base.Comment.body
- TDL::Annotation.annotatedElement := UP4TDL::Annotation.base_Comment.annotatedElement

Constraints
None.

5.9 AnnotationType

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::PrimitiveType

Generalization
None.

Properties
None.

Constraints
None.

5.10 TestObjective

Extended UML Meta-Class

- UML::Comment
- UML::Class

Generalization
None.

Properties

- <<UP4TDL::TestObjective>>
- TDL::TestObjective.description := UP4TDL::TestObjective.description
- TDL::TestObjective.objectiveURI := UP4 TDL::TestObjective.objectiveURI (new property)

Constraints
None.

6 Data

6.1 Data Definition

6.1.1 Overview

TDL Data Mapping-related concepts are mapped to stereotypes in UML.
TDL::Function concept is mapped to a stereotype to make it possible to add the constraint that a Function has exactly one return parameter.

All the other TDL Data Definition concepts are directly mapped to UML meta-classes without stereotypes.

Figure 6.1: Data Definition concepts

6.1.2 DataResourceMapping

Extended UML Meta-Class

- UML::Class

Generalization

None.

Properties

- %<UP4TDL::DataresourceMapping>>
- TDL::DataResourceMapping.resourceURI := UP4TDL::DataresourceMapping.resourceURI (new property)

Constraints

None.

6.1.3 MappableDataElement

This MetaClass shall not be mapped.

6.1.4 DataElementMapping

Extended UML Meta-Class

- UML::Dependency

Generalization

None.

Properties

- %<UP4TDL::DataElementMapping>>
- TDL::DataElementMapping.elementURI := UP4TDL::DataElementMapping.elementURI (new property)
6.1.5 ParameterMapping

Extended UML Meta-Class

- UML::Expression

Generalization

None.

Properties

- <<UP4TDL::ParameterMapping>>
- TDL::ParameterMapping.memberURI := UP4TDL::ParameterMapping.memberURI (new property)
- TDL::ParameterMapping.parameter := UP4TDL::ParameterMapping.parameter (new property)

Constraints

None.

6.1.6 DataType

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::Classifier

Generalization

None.

Properties

None.

Constraints

None.

6.1.7 DataInstance

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::InstanceSpecification
Generalization
None.

Properties
- TDL::DataInstance.dataType := UML::InstanceSpecification.classifier

Constraints
- TDL data instance shall have only one classifier

6.1.8 SimpleDataType

Extended UML Meta-Class
- Direct mapping without a stereotype to UML::PrimitiveType

Generalization
None.

Properties
None.

Constraints
None.

6.1.9 SimpleDataInstance

Extended UML Meta-Class
- Direct mapping without a stereotype to UML::InstanceSpecification

Model to Model Transformation advice
- TDL::SimpleDataInstance and TDL::StructuredDataInstance are mapped to the same UML concept: UML::InstanceSpecification. To know whether an InstanceSpecification represents a TDL::SimpleDataInstance or a TDL::StructuredDataInstance, one shall look at UML::InstanceSpecification.classifier. If it is a PrimitiveType, then the InstanceSpecification represents a TDL::SimpleDataInstance, otherwise, it represents a TDL::StructuredDataInstance
6.1.10 StructuredDataType

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::DataType

Generalization
None.

Properties

- TDL::StructuredDataType.member := UML::DataType.ownedAttribute

Constraints
None.

6.1.11 Member

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::Property

Generalization
None.

Properties
None.

Constraints
None.

Model to Model Transformation advice

- TDL::Members correspond to Properties that are owned by a DataType

6.1.12 StructuredDataInstance

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::InstanceSpecification

Generalization
None.

Properties

- TDL::StructuredDataInstance.memberAssignment:= UML::InstanceSpecification.slot

Constraints
None.
Model to Model Transformation advice

- TDL::SimpleDataInstance and TDL::StructuredDataInstance are mapped to the same UML concept: UML::InstanceSpecification. To know whether an InstanceSpecification represents a TDL::SimpleDataInstance or a TDL::StructuredDataInstance, one shall look at UML::InstanceSpecification.classifier. If it is a PrimitiveType, then the InstanceSpecification represents a TDL::SimpleDataInstance, otherwise, it represents a TDL::StructuredDataInstance.

6.1.13 MemberAssignment

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::Slot

Generalization

None.

Properties

- TDL::MemberAssignment.memberSpec := UML::Slot.value
- TDL::MemberAssignment.member := UML::Slot.definingFeature

Constraints

None.

6.1.14 Parameter

Extended UML Meta-Class

- Direct mapping without a stereotype to ConnectableElement

Generalization

None.

Properties

- TDL::Parameter.dataType := UML::ConnectableElement.type

Constraints

None.

6.1.15 FormalParameter

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::Parameter

Generalization

None.

Properties

None.
Constraints
None.

6.1.16 Variable

Extended UML Meta-Class
- Direct mapping without a stereotype to UML::Property

Generalization
None.

Properties
- TDL::Variable.dataType := UML.Property.type

Constraints
None.

6.1.17 Action

Extended UML Meta-Class
- Direct mapping without a stereotype to UML::OpaqueBehaviour

Generalization
None.

Properties
- TDL::Action.body := UML::OpaqueBehaviour.body
- TDL::Action.formalParameter := UML::OpaqueBehaviour.ownedParameter

Constraints
None.

6.1.18 Function

Extended UML Meta-Class
- UML::FunctionBehaviour

Generalization
None.

Properties
- TDL::Function.returnType := UP4TDL::Function.returnType (Derived property computed as the type of the out parameter of the underlying behaviour)
6.2 Data Use

6.2.1 Overview

Most of the Data Use concepts require a stereotype. In most of the cases these stereotypes extend the UML::Expression meta-class, except for DataUse, which extends ValueSpecification.

Figure 6.2: Data use concepts Part 1

Figure 6.3: Data use concepts Part 2

6.2.2 DataUse

Extended UML Meta-Class

- UML::ValueSpecification. This is a required extension

Generalization

None.

Properties

- 
  - TDL::DataUse.argument := UP4TDL::DataUse.argument (new property)
  - TDL::DataUse.reduction := UP4TDL::DataUse.reduction (new property)
6.2.3 ParameterBinding

Extended UML Meta-Class

- UML::Expression

Generalization

None.

Properties

- <<UP4TDL::ParameterBinding>>
- TDL::ParameterBinding.dataUse := UML::Expression.operand
- TDL::ParameterBinding.parameter := UP4TDL::ParameterBinding.parameter (new property)

Constraints

None.

6.2.4 StaticDataUse

Extended UML Meta-Class

N/A.

Generalization

N/A.

Properties

N/A.

Constraints

None.

6.2.5 DataInstanceUse

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::InstanceValue

Generalization

None.

Properties

- TDL::DataInstanceUse.dataInstance := UML::InstanceValue.instance
Constraints
None.

6.2.6 SpecialValueUse

Extended UML Meta-Class
- UML::Expression

Generalization
None.

Properties
- <<UP4TDL::SpecialValueUse>>

Constraints
None.

6.2.7 AnyValue

Extended UML Meta-Class
- UML::Expression

Generalization
- UP4TDL::SpecialValueUse

Properties
- <<UP4TDL::AnyValue>>
- TDL:: AnyValue.type := UP4TDL:: AnyValue.type (new property)

Constraints
None.

6.2.8 AnyValueOrOmit

Extended UML Meta-Class
- UML::Expression

Generalization
- UP4TDL::SpecialValueUse

Properties
- <<UP4TDL::AnyValueOrOmit>>>
6.2.9 OmitValue

Extended UML Meta-Class
- UML::Expression

Generalization
- UP4TDL::SpecialValueUse

Properties
- \texttt{\textless \textless UP4TDL::OmitValue\textgreater \textgreater}

Constraints
None.

6.2.10 DynamicDataUse

Extended UML Meta-Class
N/A.

Generalization
N/A.

Properties
N/A.

Constraints
None.

6.2.11 FunctionCall

Extended UML Meta-Class
- UML::Expression

Generalization
None.

Properties
- \texttt{\textless \textless UP4TDL::FunctionCall\textgreater \textgreater}
- TDL::FunctionCall.function := UP4TDL::FunctionCall.function (new property)

Constraints
None.
6.2.12 FormalParameterUse

Extended UML Meta-Class

- UML::Expression

Generalization

None.

Properties

- <<UP4TDL::FormalParameterUse>>
- TDL::FormalParameterUse.parameter = UP4TDL::FormalParameterUse.parameter (new property)

Constraints

None.

6.2.13 VariableUse

Extended UML Meta-Class

- UML::Expression

Generalization

None.

Properties

- <<UP4TDL::VariableUse>>
- TDL::VariableUse.variable := UP4TDL::VariableUse.variable (new property)
- TDL::VariableUse.componentInstance := UP4TDL::VariableUse.componentInstance (new property)

Constraints

None.

7 Time

7.1 Overview

All Time-related TDL concepts require stereotypes as shown in Figure 7.1 to Figure 7.3.
Figure 7.1: General time-related concepts

Figure 7.2: Time operations

Figure 7.3: Timer and Timer Operations
7.2  Time

Extended UML Meta-Class
- UML::PrimitiveType (by generalization)

Generalization
None.

Properties
- <<UP4TDL::Time>>

Constraints
None.

7.3  TimeLabel

Extended UML Meta-Class
- UML::Property

Generalization
None.

Properties
- <<UP4TDL::TimeLabel>>

Constraints
None.

7.4  TimeLabelUse

Extended UML Meta-Class
- UML::Expression (by generalization)

Generalization
None.

Properties
- <<UP4TDL::TimeLabelUse>>
- TDL::TimeLabelUse.timeLabel := UP4TDL::TimeLabelUse.timeLabel (new property)

Constraints
None.
7.5 TimeConstraint

Extended UML Meta-Class

- UML::IntervalConstraint

Generalization

None.

Properties

- <<UP4TDL::TimeConstraint>>
- TDL::TimeConstraint.timeConstraintExpression := UP4TDL::TimeConstraint.timeConstraintExpression (new property)

Constraints

None.

7.6 TimeOperation

Extended UML Meta-Class

- UML::OccurrenceSpecification

Generalization

- UP4TDL::AtomicBehaviour

Properties

- <<UP4TDL::TimeOperation>>
- TDL::TimeOperation.period := UP4TDL::TimeOperation.period (new property)
- TDL::TimeOperation.ComponentInstance := UP4TDL::ComponentInstance.base_OccurrenceSpecification.covered.represents

Constraints

None.

7.7 Wait

Extended UML Meta-Class

- UML::DurationConstraint

Generalization

- UP4TDL::TimeOperation

Properties

- << UP4TDL::Wait>>
Constraints
None.

7.8 Quiescence

Extended UML Meta-Class

- UML::DurationConstraint

Generalization

- UP4TDL::TimeOperation

Properties

- << UP4TDL::Quiescence>>
- TDL::Quiescence.gateReference := UP4TDL::Quiescence. gateReference (new property)

Constraints
None.

7.9 Timer

Extended UML Meta-Class

- UML::Interface

Generalization

None.

Properties

None.

Constraints
None.

7.10 TimerOperation

Extended UML Meta-Class

- UML::CallOperationAction

Generalization

None.

Properties

- <<UP4TDL::TimerOperation>>
- TDL::TimerOperation.timer := UP4TDL::TimerOperation.timer (new property)
- TDL::TimerOperation.componentInstance := UP4TDL::TimerOperation..componentInstance (new property)

Constraints
None.

7.11 TimerStart

Extended UML Meta-Class
- UML::CallOperationAction

Generalization
- UP4TDL::TimerOperation

Properties
- <<UP4TDL::TimerStart>>
- TDL::TimerStart.period := UP4TDL::TimerStart.period (new property)

Constraints
None.

7.12 TimerStop

Extended UML Meta-Class
- UML::CallOperationAction

Generalization
- UP4TDL::TimerOperation

Properties
- <<UP4TDL::TimerStop>>

Constraints
None.

7.13 TimeOut

Extended UML Meta-Class
- UML::CallOperationAction

Generalization
- UP4TDL::TimerOperation

Properties
- <<UP4TDL::TimeOut>>
8 Test Configuration

8.1 Overview

The TDL concepts shown on Figure 8.1 and Figure 8.2 require stereotypes in UP4TDL. The other Test Configuration-related concepts are directly mapped.

8.2 GateType

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::DataType or UML::Interface containing several datatype (for the case of multipleDataType handled by the GateInstance)

Generalization

None.

Properties

- TDL::GateType.dataType := UML::Interface.ownedAttribute.type[1..*] {unique}

Constraints

None.
8.3 GateInstance
Extended UML Meta-Class

- Direct mapping without a stereotype to UML::Port

Generalization
None.

Properties

- TDL::GateInstance.gateType := UML::Port.type

Constraints
None.

8.4 ComponentType
Extended UML Meta-Class

- UML::Class

Generalization
None.

Properties

- <UP4TDL::ComponentType>>
- TDL::ComponentType.timer:=UP4TDL::ComponentType.timer (new property)
- TDL::ComponentType.variable := UP4TDL::ComponentType.base_Class.ownedProperty
- TDL::ComponentType.gateInstance := UP4TDL::ComponentType.base_Class.ownedPorts

Constraints
None.

8.5 ComponentInstance
Extended UML Meta-Class

- UML::Property
- UML::Component

Generalization
None.

Properties

- <<UP4TDL::ComponentInstance>>
8.6 ComponentInstanceRole

Extended UML Meta-Class

- ComponentInstanceRole is modelled as an instance of a UML::Enumeration and not as a meta-class

Generalization
None.

Properties
None.

Literals

- 
  SUT
  UML::EnumerationLiteral <<SUT>>

- 
  Tester
  UML::EnumerationLiteral <<Tester>>

Constraints
None.

8.7 GateReference

The TDL::GateReference concept directly mapped to the UML::ConnectorEnd concept.

Extended UML Meta-Class

- Direct mapping to UML::ConnectorEnd

Generalization
None.

Properties

- TDL::GateReference.component := UML::ConnectorEnd.partWithPort (new property)
- TDL::GateReference.gate := UML::ConnectorEnd.role

Constraints
None.
8.8 Connection

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::Connector

Generalization

None.

Properties

- TDL::Connection.endPoint := UML::Connector.end

Constraints

None.

8.9 TestConfiguration

Extended UML Meta-Class

- UML::StructuredClassifier

Generalization

None.

Properties

- <<UP4TDL::TestConfiguration>>
- TDL::TestConfiguration.component := UP4TDL::TestConfiguration.componentInstance (new derived property computed as the subset of ownedAttribute that have stereotype componentInstance applied on)
- TDL::TestConfiguration.connector := UP4TDL::TestConfiguration.base_StructuredClassifier.ownedConnector

Constraints

None.

9 Test Behaviour

9.1 Test Description

9.1.1 Overview

TDL::TestDescription is mapped to UML::BehaviouredClassifier, while there is no need to map TDL::BehaviourDescription.
9.1.2 TestDescription

Extended UML Meta-Class

- UML::BehaviouredClassifier

Generalization

None.

Properties

- \texttt{<< UP4TDL::TestDescription>>}
- TDL::TestDescription.testConfiguration := UP4TDL::TestDescription.testConfiguration (new property)
- TDL::TestDescription.formalParameter := UP4TDL::TestDescription.parameter (new property)
- TDL::TestDescription.behaviourDescription := UP4TDL::TestDescription.base_BehaviouredClassifier.classifierBehaviour
- TDL::TestDescription.testObjective := UP4TDL::TestDescription.testObjective (new property)

Constraints

None.

9.1.3 BehaviourDescription

Extended UML Meta-Class

- Direct mapping without a stereotype to UML::InteractionFragment

Generalization

None.

Properties

None.
Constraints
None.

Model to Model transformation advice

- The BehaviourDescription is the first retrieved as the classifierBehaviour of the TestDescription

9.2 Combined Behaviour

9.2.1 Overview

Behaviour is mapped to UML::InteractionFragment, CombinedBehaviours are mapped to CombinedFragment, Blocks are mapped to UML::InteractionOperand. They are required extensions. SingleCombinedBehaviour and MultipleCombinedBehaviour are not to be mapped, but all of their sub-classes are mapped to UML::CombinedFragment.

9.2.2 Behaviour

Extended UML Meta-Class

- UML::InteractionFragment. This is a required extension

Generalization
None.
Properties
- <<UP4TDL::Behaviour>>
- TDL::Behaviour.testObjective := UP4TDL::Behaviour.testObjective

Constraints
None.

9.2.3 Block

Extended UML Meta-Class
- UML::InteractionOperand. This is a required extension

Generalization
None.

Properties
- <<UP4TDL::Block>>
- TDL::Block.behaviour := UP4TDL::Block.base_InteractionOperand.fragment
- TDL::Block.guard := UP4TDL::Block.base_InteractionOperand.guard.specification

Constraints
None.

9.2.4 CombinedBehaviour

Extended UML Meta-Class
- UML::CombinedFragment

Generalization
- UP4TDL::Behaviour

Properties
- <<UP4TDL::CombinedBehaviour>>
- TDL::CombinedBehaviour.periodic := UP4TDL::CombinedBehaviour.periodic (new property)
- TDL::CombinedBehaviour.exceptional := UP4TDL::CombinedBehaviour.exceptional (new property)

Constraints
None.

9.2.5 SingleCombinedBehaviour

This MetaClass shall not be mapped. The constraints on the number of owned blocks is transferred to the actual Combined Behaviours (i.e. CompoundBehaviour, BoundedLoopBehaviour and UnboundedLoopBehaviour). This is done in order to limit the number of stereotypes and the stereotype hierarchy length.
9.2.6 CompoundBehaviour

Extended UML Meta-Class
- UML::CombinedFragment (by generalization) with one InteractionOperand (block)

Generalization
- UP4TDL::CombinedBehaviour

Properties
- << UP4TDL::CombinedBehaviour>>

Constraints
None.

9.2.7 BoundedLoopBehaviour

Extended UML Meta-Class
- UML::CombinedFragment (by generalization) with one InteractionOperand (block)

Generalization
- UP4TDL::CombinedBehaviour

Properties
- << UP4TDL::BoundedLoopBehaviour>>
- TDL::BoundedLoopBehaviour.numIteration := UP4TDL::BoundedLoopBehaviour.numIteration (new property)

Constraints
None.

9.2.8 UnboundedLoopBehaviour

Extended UML Meta-Class
- UML::CombinedFragment (by generalization) with one InteractionOperand (block)

Generalization
- UP4TDL::CombinedBehaviour

Properties
- <<UP4TDL::UnboundedLoopBehaviour>>

Constraints
None.
9.2.9 MultipleCombinedBehaviour

This MetaClass shall not be mapped. The constraints on the number of owned blocks is transferred to the actual Combined Behaviour. This is done in order to limit the number of stereotypes and the stereotype hierarchy length.

9.2.10 AlternativeBehaviour

Extended UML Meta-Class

- UML::CombinedFragment (by generalization) with at least 2 InteractionOperand (block)

Generalization

- UP4TDL::CombinedBehaviour

Properties

- <<UP4TDL::AlternativeBehaviour>>

Constraints

None.

9.2.11 ConditionalBehaviour

Extended UML Meta-Class

- UML::CombinedFragment (by generalization) with one or moreInteractionOperand (block)

Generalization

- UP4TDL::CombinedBehaviour

Properties

- <<UP4TDL::ConditionalBehaviour>>

Constraints

None.

9.2.12 ParallelBehaviour

Extended UML Meta-Class

- UML::CombinedFragment (by generalization) with one or moreInteractionOperand (block)

Generalization

- UP4TDL::CombinedBehaviour

Properties

- <<UP4TDL::ParallelBehaviour>>
Constraints
None.

9.2.13 ExceptionalBehaviour

Extended UML Meta-Class

- UML::CombinedFragment (by generalization) with one InteractionOperand (block)

Generalization

- UP4TDL::CombinedBehaviour

Properties

- <<UP4TDL::ExceptionalBehaviour>>
- TDL:: ExceptionalBehaviour.block := UP4TDL:: ExceptionalBehaviour.base_CombinedFragment.operand
- TDL:: ExceptionalBehaviour.guardedComponent := UP4TDL::ExceptionalBehaviour.guardedComponent (new property)

Constraints
None.

9.2.14 DefaultBehaviour

Extended UML Meta-Class

- UML::CombinedFragment (by generalization)

Generalization

- UP4TDL::ExceptionalBehaviour

Properties

- << UP4TDL::DefaultBehaviour>>

Constraints
None.

9.2.15 InterruptBehaviour

Extended UML Meta-Class

- UML::CombinedFragment (by generalization)

Generalization

- UP4TDL::ExceptionalBehaviour

Properties

- << UP4TDL::InterruptBehaviour>>
Constraints
None.

9.2.16 PeriodicBehaviour

Extended UML Meta-Class
- UML::CombinedFragment (by generalization)

Generalization
- UP4TDL::CombinedBehaviour

Properties
- &lt;&lt; UP4TDL::PeriodicBehaviour&gt;&gt;
- TDL:: PeriodicBehaviour.block := UP4TDL::PeriodicBehaviour.base_CombinedFragment.operand
- TDL:: PeriodicBehaviour.period := UP4TDL:: PeriodicBehaviour (new property)

Constraints
None.

9.3 Atomic Behaviour

9.3.1 Overview
All TDL::AtomicBehaviour-related concepts require stereotypes, as shown in Figure 9.4 to Figure 9.7.

Figure 9.4: Global Atomic Behaviour concepts
9.3.2 AtomicBehaviour

Extended UML Meta-Class
- UML::InteractionFragment

Generalization
None.

Properties
- \texttt{<<UP4TDL::AtomicBehaviour>>}
- \texttt{TDL::AtomicBehaviour.timeLabel} := \texttt{UP4TDL::AtomicBehaviour.timeLabel} (new property)
- \texttt{TDL::AtomicBehaviour.timeConstraint} := \texttt{UP4TDL::AtomicBehaviour.timeConstraint} (new property)

Constraints
None.

9.3.3 Break

Extended UML Meta-Class
- UML::OccurrenceSpecification

Generalization
- \texttt{UP4TDL::AtomicBehaviour}

Properties
- \texttt{<< UP4TDL::Break>>}

Constraints
None.

9.3.4 Stop

Extended UML Meta-Class
- UML::OccurrenceSpecification

Generalization
- \texttt{UP4TDL::AtomicBehaviour}

Properties
- \texttt{<< UP4TDL::Stop>>}

Constraints
None.
9.3.5 VerdictAssignment

Extended UML Meta-Class
- UML::OccurrenceSpecification
- UML::StateInvariant

Generalization
- UP4TDL::AtomicBehaviour

Properties
- << UP4TDL::VerdictAssignment>>
- TDL::VerdictAssignment.verdict := UP4TDL::VerdictAssignment.verdict

Constraints
None.

9.3.6 Assertion

Extended UML Meta-Class
- UML::OccurrenceSpecification
- UML::StateInvariant

Generalization
- UP4TDL::AtomicBehaviour

Properties
- << UP4TDL::Assertion>>
- TDL::Assertion.condition := UP4TDL::Assertion.condition (new property)
- TDL::Assertion.otherwise := UP4TDL::Assertion.otherwise (new property)

Constraints
None.

9.3.7 Interaction

Extended UML Meta-Class
- UML::Message

Generalization
- UP4TDL::AtomicBehaviour

Properties
- <<UP4TDL::Interaction>>
- TDL::Interaction.isTrigger := UP4TDL::Interaction.isTrigger (new property)
- TDL::Interaction.argument := UP4TDL::Interaction.base_Message.argument
- TDL::Interaction.target := UP4TDL::Interaction.targets ; (the first is UP4TDL::Interaction.base_Message.receiveEvent)
- UP4TDL::Interaction contains a set of connector UP4TDL::Interaction.connection (new derived property computed as the union of all target connectors of the interactions) that allows the derivation of the source gate and the target gates
- TDL::Interaction.sourceGate : UP4TDL::Interaction.sourceGate (new derived property computed from as the single sending Port)
- TDL::Interaction.targetGate := UP4TDL::Interaction.targetGate (new derived property computed as the union of all receiving Ports)

Constraints
None.

9.3.8 Target

Extended UML Meta-Class
- UML::(Message)OccurenceSpecification

Generalization
None.

Properties
- <<UP4TDL::Target>>
- TDL::Target.variable := UP4TDL::Target.variable
- TDL::Target.targetGate := UP4TDL::Target retrieved from UP4TDL::Target.connection (see Model to model Transformation advice)

Constraints
None.

Model to model Transformation Advice
- UP4TDL::Target contains a property connection of type UML::Connector. TDL::Target.targetGate t is the Retrieved from a UP4TDL::Target as follow:
  - It is the end of UP4TDL::Target.end ce such that : ce.partWithPort is the same as t.covered.represent.

9.3.9 TestDescriptionReference

Extended UML Meta-Class
- UML::InteractionUse

Generalization
None.
Properties

- \(<\text{UP4TDL::TestDescriptionReference}>\)
- \(\text{TDL::TestDescriptionReference.testDescription} := \text{UP4TDL::TestDescriptionReference.base_InteractionUse.referTo}\)
- \(\text{TDL::TestDescriptionReference.actualParameter} := \text{UP4TDL::TestDescriptionReference.base_InteractionUse.argument}\)
- \(\text{TDL::TestDescriptionReference.componentInstanceBinding} := \text{UP4TDL::TestDescriptionReference.componentInstanceBinding (new property)}\)

Constraints

None.

9.3.10 ComponentInstanceBinding

Extended UML Meta-Class

- \(\text{UML::Class}\)

Generalization

None.

Properties

- \(<\text{UP4TDL::ComponentInstanceBinding}>\)
- \(\text{TDL::ComponentInstanceBinding.formalComponent} := \text{UP4TDL::ComponentInstanceBinding.formalComponent (new property)}\)
- \(\text{TDL::ComponentInstanceBinding.actualComponent} := \text{UP4TDL::ComponentInstanceBinding.actualComponent (new property)}\)

Constraints

None.

9.3.11 ActionBehaviour

Extended UML Meta-Class

- \(\text{UML::InstanceFragment (by generalization)}\)

Generalization

- \(\text{UP4TDL::AtomicBehaviour}\)

Properties

- \(<\text{UP4TDL::ActionBehaviour}>\)
- \(\text{TDL::ActionBehaviour.componentInstance} := \text{retrieved from the lifeline that this UP4TDL::ActionBehaviour covers. (see Model to model Transformation advice)}\)
Constraints
None.

Model to model transformation advice
- The componentInstance is retrieved from the child concepts
- For ActionReference:
  - TDL::ActionBehaviour.componentInstance = UP4TDL::ActionReference.base_BehaviourExecutionSpecification.start.covered.represents
- For InlineAction:
  - TDL::ActionBehaviour.componentInstance = InlineAction.base_OccurrenceSpecification.covered.represents
- For Assignment:
  - TDL::ActionBehaviour.componentInstance = Assignment.base_OccurrenceSpecification.covered.represents

9.3.12 ActionReference

Extended UML Meta-Class
- UML::BehaviourExecutionSpecification
- UML::ActionExecutionSpecification

Generalization
- UP4TDL::ActionBehaviour

Properties
- << UP4TDL::ActionReference >>
- TDL::ActionReference.action := UP4TDL::ActionReference.base_BehaviourExecutionSpecification.behaviour
- TDL::ActionReference.actualParameter := UP4TDL::ActionReference.actualParameter (new property)

Constraints
None.

9.3.13 InlineAction

Extended UML Meta-Class
- UML::OccurrenceSpecification

Generalization
- UP4TDL::ActionBehaviour

Properties
- <<UP4TDL::InlineAction>>
- TDL::InlineAction.body := UP4TDL::InlineAction.body (new property)

Constraints
None.

9.3.14 Assignment

Extended UML Meta-Class
- UML::OccurrenceSpecification

Generalization
- UP4TDL::ActionBehaviour

Properties
- <<UP4TDL::Assignment>>
- TDL::Assignment.variable := UP4TDL::Assignment.variable
- TDL::Assignment.expression := UP4TDL::Assignment.expression

Constraints
None.
### History

<table>
<thead>
<tr>
<th>Document history</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.1.1</td>
</tr>
<tr>
<td>V1.1.1</td>
</tr>
</tbody>
</table>
